

T1.4.7 - Investigación en herramienta de valoración de la reducción de emisiones del empleo de la computación cuántica

Introducción	1
Diseño de la herramienta qilisensors	1
Implementación de la herramienta en un entorno real	2
Implementación en laboratorio	4
Propuesta de benchmarking	4

1. Introducción

El fin de esta tarea es monitorizar, analizar y comparar, mediante benchmarks, el nivel de emisiones de un ordenador cuántico frente a sistemas de computación clásicos. Debido a la inexistencia de una herramienta que permita evaluar el consumo de un laboratorio de computación cuántica diseñamos y desarrollamos un sistema de sensores y monitorización remota que llamaremos qilisensors. Con la instalación de esta herramienta pasamos a monitorizar una serie de benchmarks o algoritmos con un conjunto de métricas seleccionadas y las comparamos con la ejecución de la misma tarea en un sistema de supercomputación convencional. Investigamos la relación que tiene nuestro conjunto de métricas con el nivel de emisiones de CO₂ y con ello comparamos ambos paradigmas computacionales.

2. Diseño de la herramienta qilisensors

Pasamos a elaborar el esquema del diseño de la herramienta de monitorización. El esquema de la herramienta consiste en un conjunto de sensores IOT de medición de consumo eléctrico instalados sobre cada uno de los dispositivos electrónicos. Dichos sensores envían datos de manera continua a una base de datos específica para series temporales (InfluxDB) mediante el protocolo de transferencia de datos IOT MQTT. La base de datos que recoge las mediciones se conecta directamente con un sistema de visualización interactiva (Grafana) que permite al usuario acceder y analizar los datos de manera online.

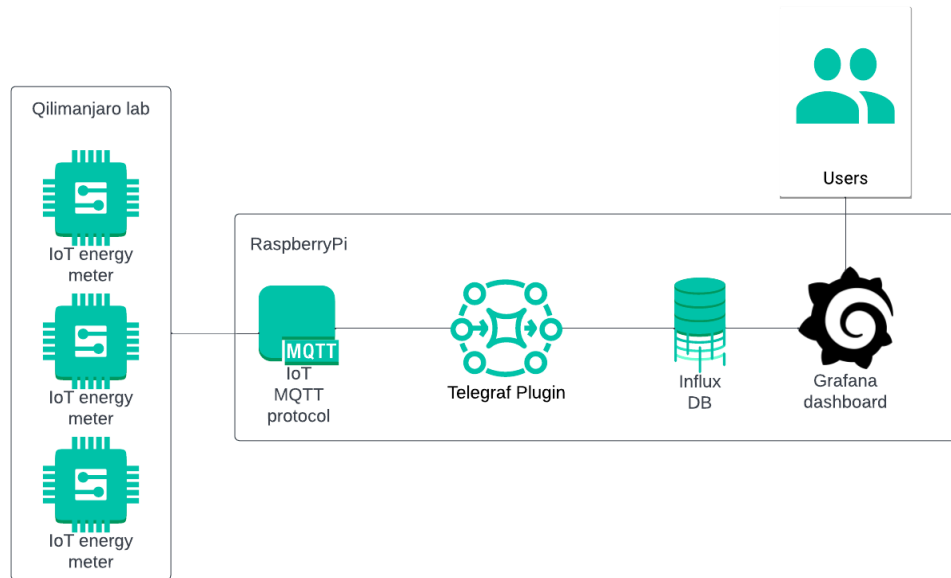


Fig 1. Diagrama de qilisensors con los elementos y protocolos implicados.

3. Implementación de la herramienta en un entorno real

Una vez terminado el diseño pasamos a programar la herramienta y implementarla en un entorno real. Para ello adquirimos un sensor de energía de tipo pinza (<https://shelly-api-docs.shelly.cloud/gen1/#shelly-em-overview>) y lo instalamos en un cuadro eléctrico casero. Llevamos a cabo los siguientes pasos:

- A. Conectar el sensor a la red wifi local
 - a. Acceder a la red wifi creada por el dispositivo 'Shelly-XXXXX'
 - b. En el navegador acceder a 192.168.33.1
 - c. Configurar la red WiFi del dispositivo a la red local
 - d. Obtener la nueva IP del dispositivo
- B. Acceder a la interfaz de configuración de Shelly (acceder a su nueva IP en el navegador) y configurar el envío de datos vía protocolo mqtt a la IP y puerto del servidor.
- C. Con el fin de simplificar el despliegue de la herramienta en otros dispositivos y entornos decidimos contenerizar los servicios usando Docker.
 - a. Creamos un archivo docker-compose (v3.8) con una red propia y los siguientes servicios
 - i. mosquitto
 - ii. influxdb

- iii. telegraf
- iv. grafana
- b. Configuramos el mapeo de puertos para garantizar que todo esté bien comunicado

```
docker-compose.yml
1  version: '3.8'
2
3  services:
4
5      mosquito:
6          image: eclipse-mosquitto:latest
7          restart: always
8          ports:
9              - "1881:1881"
10             - "9001:9001"
11          volumes:
12              - ./mosquitto/mosquitto.conf:/mosquitto/config/mosquitto.conf
13          networks:
14              - emeter
15
16      influxdb:
17          image: influxdb:1.8.10
18          restart: always
19          ports:
20              - "8086:8086"
21          networks:
22              - emeter
23          volumes:
24              - influxdb-data:/var/lib/influxdb
25
26      telegraf:
27          image: telegraf
28          restart: always
29          volumes:
30              - ./telegraf.conf:/etc/telegraf/telegraf.conf:ro
31          depends_on:
32              - mosquito
33              - influxdb
34          networks:
35              - emeter
36
37      grafana:
38          image: grafana/grafana
39          restart: always
40          ports:
41              - "3000:3000"
42          networks:
43              - emeter
44          volumes:
45              - grafana-data:/var/lib/grafana
46          depends_on:
47              - influxdb
48
49  networks:
50      emeter:
51
52  volumes:
53      influxdb-data:
54      grafana-data:
```

Fig 2. Docker-compose file

D. Creamos los archivos de configuración:

- a. mosquitto.conf: para permitir la posibilidad de incluir autenticación

```
mosquitto > ⚙ mosquitto.conf
1  persistence false
2  listener 1881
3  log_dest stdout
4  allow_anonymous true
5  connection_messages true
6
```

Fig 3. Archivo de configuración de protocolo mqtt

- b. telegraf.conf: para definir los topics a los que se tiene que suscribir el broker mqtt, el formato de los datos a recibir y la base de datos donde van a ir alojados ('emeter').

```
⚙ telegraf.conf
1  [[outputs.influxdb]]
2    urls = ["http://influxdb:8086"]
3    database = "emeter"
4
5
6  [[inputs.mqtt_consumer]]
7    servers = ["tcp://mosquitto:1881"]
8    topics = [
9      "shellies/shellyem-BCFF4DFD0908/emeter/0/energy",
10     "shellies/shellyem-BCFF4DFD0908/emeter/0/total",
11     "shellies/shellyem-BCFF4DFD0908/emeter/0/power",
12     "shellies/shellyem-BCFF4DFD0908/emeter/0/voltage",
13     "shellies/shellyem-BCFF4DFD0908/emeter/0/reactive_power"
14   ]
15   data_format = "value"
16   data_type = "float"
```

Fig 4. Archivo de configuración del plugin telegraf

- E. Una vez configurado y ejecutado 'docker-compose up -d' ya tenemos todos los servicios necesarios levantados y recogiendo datos del sensor.
- F. En el navegador abrimos un servicio de grafana asociado al puerto local definido en el docker file (<http://localhost:3000>) y configuramos la fuente de datos como la base de datos InfluxDB creada ('emeter'), seleccionamos las queries para obtener los datos que nos interesan y montamos el dashboard (visualización) de los mismos.
- G. Las métricas recogidas por unidad de tiempo (cada minuto) son las siguientes:

- a. Energía
- b. Potencia activa
- c. Potencia reactiva
- d. Voltage



Fig 5. Visualización de la monitorización del consumo energético en tiempo real mediante el dashboard de Grafana.

4. Propuesta de benchmarking

Para hacer la comparativa entre sistemas computacionales clásicos y cuánticos, seleccionamos un algoritmo representativo para ejecutar durante el experimento y el proceso de monitorización. Debido a su versatilidad y fácil escalamiento se ha elegido el algoritmo de *randomized benchmarking* para varias profundidades de circuito. Este algoritmo consiste en la aplicación de varias secuencias de puertas lógicas cuánticas de Clifford seleccionadas de manera aleatoria. Se ha implementado el algoritmo dentro del framework de programación y control propio de Qilimanjaro: Qibo.

El protocolo de randomized benchmarking que hemos elegido se usa generalmente para evaluar la calidad de las puertas y del control de un dispositivo cuántico. Se basa en dos propiedades del grupo de Clifford:

- la secuencia de operaciones Clifford equivale a una sola operación de Clifford y
- la inversa de una operación de Clifford también forma parte del mismo grupo.

Por lo tanto podemos describir nuestro algoritmo con los siguientes pasos:

1. Inicializamos el qubit en el ground state
2. Aplicamos n puertas de Clifford consecutivas
3. Aplicamos una puerta de Clifford final que invierte todas las anteriores
4. Medimos el estado del qubit

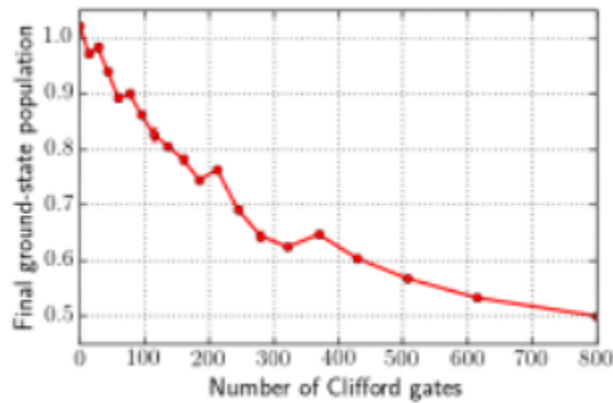


Fig 6. Simulación de un proceso típico de randomized benchmarking

5. Implementación en laboratorio

Las máquinas que se usarán para las ejecuciones de dichos tests algorítmicos són los dispositivos cuánticos basados en qubits superconductores de los dos laboratorios locales de Qilimanjaro con características muy distintas.

- Transmon qubit chip lab:



Fig 7. Laboratorio de qubits superconductores tipo transmon

- Flux qubit chip lab:

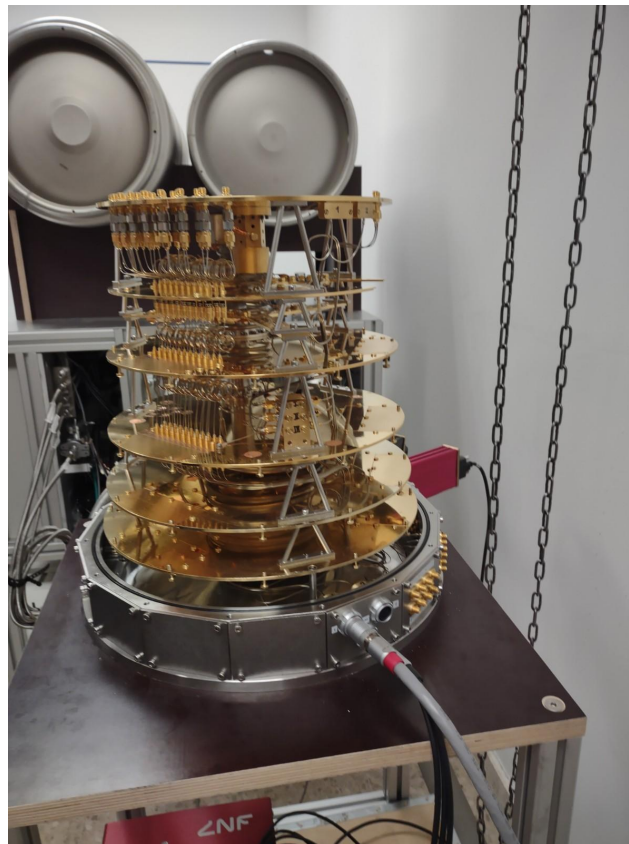


Fig 8. Laboratorio de qubits superconductores tipo flux

Identificamos todo el equipo implicado en un experimento de computación cuántica con el fin de desplegar el número adecuado de sensores. Para ello preparamos un diagrama de wiring para tener una idea global de la arquitectura de la plataforma cuántica.

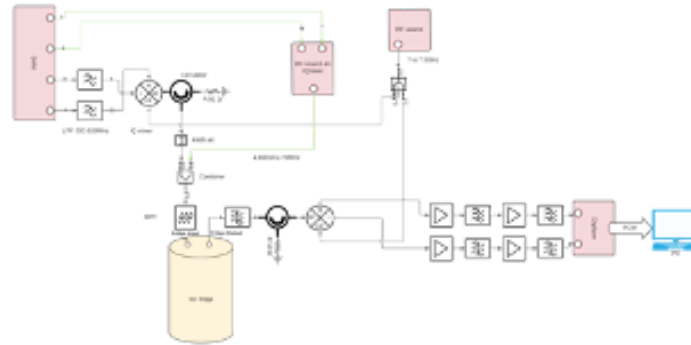


Fig 9. Diagrama de wiring del laboratorio de transmon qubit chip

Identificamos entonces los dispositivos electrónicos disponibles y necesarios para llevar a cabo un experimento con un solo qubit superconductor, su voltaje requerido y su variabilidad con número de qubits con el fin de poder tener en cuenta consideraciones de escalado en nuestro análisis.

Dispositivo	Voltaje (V)	Variabilidad
Compresor He	380	-
Chiller agua	380	-
Compresor aire	380	-
GHS	380	-
Pulsar readout	220	Si
Pulsar control	220	Si
Rhode&Schwarz QRM	220	Si

Rhode&Schwarz QCM	220	Si
Amplifiers x 2	220	Si
Amplifiers power supply x 2	220	Si
Bluefors control	220	-
Ordenador	220	-
Monitor	220	-
UPS	220	-
Router	220	-
Switch	220	-

La siguiente fase consiste en la instalación de la herramienta desarrollada con el fin de preparar el laboratorio para las tareas de implementación de los benchmarks de este mismo proyecto (T 3.7).